### Use Case Name:

Blocked Order Release Recommendation

### Use Case (internal) Alias:

BORR

### Consumer Product:

Credit Cloud

### Usual re-training Frequency:

once a month or less frequently (based on deterioration in prediction accuracy)

### Granularity of model/Feature List:

* Model trained separately for each account with respective account data
* List of Features varies across accounts

### Business Metrics Measurement Status:

Measured:

* % of recommendations followed by analyst

Unable to measure:

* Average time taken to release a Blocked Order

### Input to the AI Model for Training:

At training time:

* Closed + Open AR Data
* Closed + Open P2P Data(if available)
* Closed Orders
* Customer Master

### Data Requirements:

* Blocked Order History
* Credit data at the time of every blocked order and current credit data
* Payment History
* Payment commitment data
* AR data (closed+open invoices)
* Credit limit change history
* Customer notes
* Credit utilization history

1. Closed and open invoices data:
2. Invoice date
3. Invoice clearing date
4. Payment terms
5. Discount cash
6. Discount percent
7. Invoice due date
8. Open amount
9. No.of invoice transactions
10. Payment Commitment Data
11. Clearing date
12. Due date
13. Committed amount
14. Paid date
15. Paid amount
16. Order and credit Data
17. Order amount
18. Order create date
19. Order release date
20. Order company code
21. Open ar amount
22. Credit risk class
23. Available current credit
24. Credit used
25. Credit Score
26. Credit limit update time
27. Approved credit limit

### Features Documentation:

Invoked for a blocked order created in real time.

### Execution granularity:

Invoked for a blocked order created in real time.

### Typical end to end Execution time:

0.86 Sec

### Input at Execution:

Blocked Order created at the system

### What's returned after Execution:

For each blocked order:

* whether to unblock or not
* accompanying action like taking collateral/increasing credit limit etc)
* reason for that recommendation.

### Coverage, Exclusions:

No Exclusions.

Factors Affecting Decision:

1. Permanent credit limit increase
2. Temporary credit limit increase
3. Promise to pay
4. Payment received
5. Past payment behaviour
6. Revenue from the customer
7. Seasonal Utilization
8. Overshoot percent
9. Amount past due and aging bucket
10. Projected utilization
11. Risk class
12. Credit score and Order Aberration

Key Features:

1. Committed amount
2. No.of commitments
3. Kept ratio
4. Credit score
5. Avg delay
6. Avg gap
7. Avg days to pay
8. Avg revenue
9. Total amount past due
10. Delayed avg
11. Cc avg delay
12. Credit limit increase
13. Risk class
14. Payment received percentage
15. Utilization change
16. Overshoot with order

### Model(s) used:

* Random Forest (regression)
* LightGBM

One of these models is finalized for a given account by DS engineer during Enablement

### Pitfalls Addressed/Covered:

No Pitfalls.

### Challenges:

* For pre-golive clients, format/structure/functionality (eg derived fields) inconsistency across multiple iterations of historical data provided by the client
* Different order release criteria across client side (this is factored in feature list determination at enablement time)

### Accuracies obtained:

85-92% (Test Accuracy)

### Developers working:

Shivendra, Harkirat

### Start Month, Yr:

Aug 2020

Babba's DS team's Responsibility:

Not Applicable

Periodic Monitoring Effort:

Once every month

Type of Result Verification/accuracy measurement available:

1. Performance Monitoring Deck is prepared for all of the accounts on a monthly basis

Link to the Above Verification:

[CRD | AI | BOSE | Monthly Governance | JUNE -2021](https://docs.google.com/presentation/d/1VUcZAMFkLHqRJSSSBiRLDzsz9JEr0XwOByAI8QhM0zQ/edit?usp=sharing)

### Best Accuracy in the Above Verification:

0.49 R2 Score for **Bose**

### Model Metrics:

* Precision = True Positives/(True Positives + False Positives)
* Recall =True Positives /(True Positives + False Negatives)